|  |  |  |
| --- | --- | --- |
| Method Name & return type | Description | How it works |
| SetBackground ()  Return type: void | sets the background colour of screen | Set colour to (32,178,170)  Draw rectangle with x = 0, y =0, width = getWidth () – 1, height getHeight () - 1 |
| drawSymbol\_0 (int x, int y)  Return type: void | draws the symbol 'o' on the screen by taking co-ordinates of upper left corner as x and y value | Set background colour  Fill square using parameters (x, y) and 180 length  Set colour for symbol ‘o’  Use for loop (1 to 10), and create arc with increasing angle of i\* 36  and use Thread.sleep() to create animation |
| drawSymbol\_x (int x, int y)  Return type: void | draws the symbol 'x' on the screen by taking co-ordinates of upper left corner as x and y value | Set background colour  Fill square using parameters (x, y) and 180 length  Set colour for symbol ‘x’  Draw ovals for edges of x  Use for loop and draw small rectangle with x value x + 13 + i and y + 24 + i |
| introPage ()  Return type: void | displays the title of the game and asks user to press any key to continue | Set background  Set text colour, font size and font type  Draw tic tac toe using drawString method  Use drawSymbol\_0 and drawSymbol\_x to draw symbols on title page  Use drawString and getChar method to ask user to press any key  Clear the console using c.clear method |
| gameType ()  Return type: int | asks the player for intensity of game(easy/medium/hard)  and returns 1 for easy, 2 for medium and 3 for hard | Set background  Set text colour, font size and font type  Display game types (easy, medium, hard)and use getChar method to take input from player about game type  Convert char into integer  Check if player entered valid value, if not do the process again using while loop  Return integer |
| drawBoard ()  Return type: void | draws the tic tac toe board with animation | Set Background  Set colour for tic tac toe board  Use drawline function, for loop and Thread.sleep to draw small lines in animation to crate tic tac toe board  Display square numbers from 1 to 9 using drawString method |
| firstTurn ()  Return type: int | asks the player if he/she wants to go first and returns 1 if player wants to go first or returns 2 if player does not want to go first | Set text colour, font size and font type  Ask user at the bottom of the screen if they want to go first using getChar method  Check if it valid or not (any other char than y or n)  use while loop to take input until user input the valid value  Convert input into int data type (1 or 2)  Return integer |
| clearInput ()  Return type: void | clears the input text at the bottom of the screen | Set back ground colour  Fill rectangle at the bottom of the screen to clear input area |
| playerMove ()  Return type: int[] | playerMove takes the move from player and returns array with 2 elements first element represents row number and second element represents column number | Clear input area  Set text colour  Take square number from user as char using getChar method  Convert into integer  Check if input is valid or not  use while loop to take input until user input the valid value  convert in to row and column using convertToPos (make array to save row and column number)  check if position is empty or not, use while loop until player chooses empty square number (use board array to check if row and column is empty or not)  save the position in to board array  draw symbol x on board  return row and column number as an array |
| convertToPos (**int** boxNum)  Return type: int[] | takes the box number and convert into row and column number. It returns array with 2 elements; first element represents row number and second element represents column number | Declare a pos array (size 2, represents row and col number  Use if – else – if statement or switch and case statement to go through every possible box number (1 to 9) and find row and col number according to box number(ex box number 6 has the value of row 2 and col 1) |
| changeTurn (**int** t)  Return type: int | changeTurn takes turn value and changes the turn. It returns 1 if previous turn was 2  and returns 2 if previous turn was 1 | if (t == 1)  return 2;  else  return 1; |
| comRandomMove ()  Return type: int[] | decides to move for computer in easy mode It  returns an integer array representing values of row and column | generate random position by generating random values of row and column  check if position is empty  if yes save the move  else generate random number until you get empty position (while loop)  crate array that contain row and column number  display symbol ‘o’ on the screen  return array |
| makeOptimalMove ()  Return Type: int[] | makes random move for easy level It returns an integer array representing values of row and column | create array that contain row and column number  invoke checkComWin method and create an array with win information of computer index 0 - next move can result in win(0 if can not win, 1 if can win) index 1,2 -position of that move  invoke checkPlayerWin method and create array with win information of player index 0 - next move can result in win (0 if can not win, 1 if can win) index 1,2 -position of that move  check if computer can win (if yes store the move)  if no, check for player win (if yes store the block move)  if no, generate random number (follow procedure as random move)  draw symbol  return position array |
| makePerfectMove ()  Return Type: int[] | decides to move for computer in hard mode It  returns an integer array representing values of row and column | Follow same procedure as optimal Move method, but instead of generating random number at last, do following procedure before and then generate random number  If player’s previous move was corner take opposite corner (if empty)  Take center if empty  Then take empty corners |
| checkComWin ()  Return Type: int[] | uses the board array and finds if there are any possible move that the computer can make to  win. Then it returns an array with the information including where computer can place the move to win. | create an array with win information index 0 - next move can result in win(0 if can not win, 1 if can win index 1, 2 - value of possible win position  loop through every row number and check two in a row for symbol 0  if did not find any move, loop through every column number and check two in a row for symbol 0  if did not find any move, check two diagonals to find two in a row for symbol 0  save the winning move if find any and change first index to 1  return array |
| checkPlayerWin ()  Return Type: int[] | uses the board array and finds if there are any possible moves for the player that he/she may make and win. Then it returns an array with the information including where player will place the move to win. | Follow the same procedure as checkComWin method but instead of checking for symbol 0 check for symbol x |
| gameOverChecker (int[] pos)  Return type: boolean | checks if the game is over and returns a type Boolean (a type Boolean (true or false)) | Take the array and save the array value into two integer row and col  Check row, column and diagonal corresponding to value of row and col to find three same symbols in a row  If find three in a row return true else return false |
| tieChecker ()  Return type: boolean | checks if the game is tie or not and returns a type Boolean (if yes then true or if no then false) | Declare boolean variable tie to true  Use for loop to check if every position is fill with x and 0  If every position is filled with x or o  Set tie to false  Return tie |
| clearBoard ()  Return type: void | clears the tic tac toe board | Set back ground colour  fill square on tic tac toe board area |
| displayResult (**char** symbol)  Return type: void | displays the result of the game | Use setBackground method to clear area  Set font colour, font type and font size  if (symbol == ‘x’)  draw x wins  else if (symbol == ‘0’)  draw o wins  else  draw tie game  display press any key to continue  and use getChar method to wait until user press any key |
| playAgain ()  Return type: boolean | asks the player if he/she wants to play again returns a boolean value according to player's input. | clear input area using clearInput  clear the tic tac toe board using *clearBoard*  set font colour, font type and font size  ask user if they want to play again using getChar method  check if it is valid or not (use while loop)  return Boolean value according to char value of y and n |
| exitScreen ()  Return type: void | displays thank for playing sign on screen | set background colour  set text colour, font type and font size  draw “Thanks For Playing” using drawString method |